PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-244557
(43)Date of publication of application : 30.08.2002

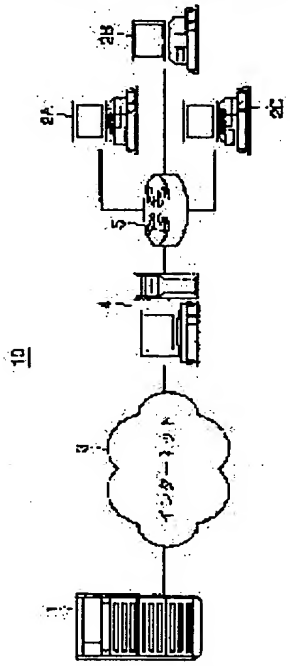
(51)Int.Cl. G09C 1/00
H04L 9/32

(21)Application number : 2001-039572
(22)Date of filing : 16.02.2001

(71)Applicant : ATR ADAPTIVE COMMUNICATIONS RES LAB
(72)Inventor : KIRIMOTO NAOKI
YAMAZAKI TATSUYA

(54) CRYPTOGRAPHIC COMMUNICATION SYSTEM AND AUTHENTICATION METHOD USED THEREFOR

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a cryptographic communication system permitting mutual authentication without transmitting own certificates to a server from clients, and to provide an authentication method used therefor.
SOLUTION: The cryptographic communication system 10 comprises a server 1, client servers 2A, 2B, 2C, the Internet network 3, CA Proxy(Certificate Authority Proxy) 4, and a coupler 5. The CA Proxy 4 is arranged between the Internet network 4 and the coupler 5. In response to a request for submittal of the digital certificates of the client servers 2A, 2B, 2c from the server 1, the CA Proxy 4 transmits to the server 1 their own digital certificates presenting that the client servers 2A, 2B, 2C are authorized servers, irrespective of whether or not the client servers 2A, 2B, 2C possess the digital certificates.



LEGAL STATUS

[Date of request for examination] 09.03.2001
[Date of sending the examiner's decision of rejection] 24.08.2004
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－244557
（P2002－244557A）
(43)公開日　平成14年８月30日(2002.8.30)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード（参考）
Ｇ０９Ｃ　1/00	６４０	Ｇ０９Ｃ　1/00	６４０Ｚ	５Ｊ１０４
Ｈ０４Ｌ　9/32		Ｈ０４Ｌ　9/00	６７５Ｄ	

審査請求　有　　請求項の数９　ＯＬ　（全 18 頁）

(21)出願番号　特願2001－39572(P2001－39572)

(22)出願日　平成13年２月16日(2001.2.16)

(71)出願人　396011680
株式会社エイ・ティ・アール環境適応通信研究所
京都府相楽郡精華町光台二丁目２番地２

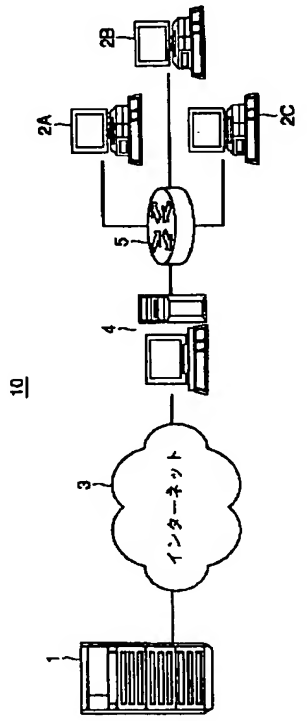
(72)発明者　桐本　直樹
京都府相楽郡精華町光台二丁目２番地２
株式会社エイ・ティ・アール環境適応通信研究所内

(74)代理人　100064746
弁理士　深見　久郎　（外４名）

最終頁に続く

(54)【発明の名称】　暗号通信システムおよびそれに用いる認証方法

(57)【要約】
【課題】　クライアントがサーバに対して自己の証明書を送信しなくても、相互認証が可能な暗号通信システムおよびそれに用いる認証方法を提供する。
【解決手段】　暗号通信システム１０は、サーバ１と、クライアントサーバ２Ａ，２Ｂ，２Ｃと、インターネット網３と、ＣＡ　Ｐｒｏｘｙ（Ｃｅｒｔｉｆｉｃａｔｅ　Ａｕｔｈｏｒｉｔｙ　Ｐｒｏｘｙ）４と、結合器５とを備える。ＣＡＰｒｏｘｙ４は、インターネット網３と結合器５との間に配置される。ＣＡ　Ｐｒｏｘｙ４は、サーバ１からクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書の提出要求に応じて、クライアントサーバ２Ａ，２Ｂ，２Ｃが電子証明書を保持しているか否かに拘わらず、サーバ１に対してクライアントサーバ２Ａ，２Ｂ，２Ｃが正規のサーバであることを示す自己の電子証明書をサーバ１へ送信する。



1

【特許請求の範囲】

【請求項１】　複数の層構造から成るプロトコルを用いて、所定の暗号方式によって暗号化された暗号データの通信を行なう暗号通信システムであって、

データまたは前記暗号データを送受信する第１のサーバと、

前記第１のサーバとの間で前記データまたは暗号データを送受信する第２のサーバと、

前記第１のサーバと前記第２のサーバとの通信を中継する第３のサーバとを備え、

前記第２のサーバの認証時、前記第３のサーバは、前記第１のサーバからの前記第２のサーバの証明書の要求に応じて、前記第２のサーバが正規のサーバであることを示す代理証明書を前記第２のサーバに代わって前記第１のサーバへ送信する、暗号通信システム。

【請求項２】　前記第３のサーバは、自己の証明書を前記代理証明書として前記第１のサーバへ送信する、請求項１に記載の暗号通信システム。

【請求項３】　前記第３のサーバは、前記第２のサーバの証明書の要求を前記第２のサーバへ送信し、前記第２のサーバから前記第２のサーバの証明書を受信すると前記代理証明書を前記第１のサーバへ送信する、請求項１または請求項２に記載の暗号通信システム。

【請求項４】　前記第３のサーバは、前記第２のサーバの証明書の要求を前記第２のサーバへ送信し、前記第２のサーバが証明書を保持しないことを示す情報を前記第２のサーバから受信すると前記代理証明書を前記第１のサーバへ送信する、請求項１または請求項２に記載の暗号通信システム。

【請求項５】　前記第３のサーバは、前記第１のサーバから受信した前記第１のサーバの証明書を証明書廃棄リストと照合し、前記第１のサーバの証明書が廃棄されているとき前記第１のサーバの証明書が無効であることを示す情報を前記第１のサーバへ送信する、請求項１から請求項４のいずれか１項に記載の暗号通信システム。

【請求項６】　前記第３のサーバは、前記第１のサーバから受信した前記第１のサーバの証明書を証明書廃棄リストと照合し、前記第１のサーバの証明書が廃棄されていないことを確認すると前記代理証明書を前記第１のサーバへ送信する、請求項１から請求項４のいずれか１項に記載の暗号通信システム。

【請求項７】　前記第３のサーバは、

前記第１のサーバと前記第２のサーバとの間の通信を制御する通信制御部と、

前記通信制御部を介して受取った前記第１のサーバの証明書を前記証明書廃棄リストと照合する証明書照合部と、

前記証明書照合部からの照合結果に基づいて、前記第１のサーバを認証し、または前記第１のサーバの証明書を無効と判定する認証部と、

2

前記通信制御部を介して前記第２のサーバの証明書または前記第２のサーバが証明書を保持しない情報を受取ると、前記代理証明書を前記第１のサーバへ送信する代理応答部とを含む、請求項５または請求項６に記載の暗号通信システム。

【請求項８】　第１のサーバと第２のサーバとの間における認証方法であって、

前記第２のサーバの証明書の送信要求を前記第１のサーバから受信する第１のステップと、

前記第２のサーバの証明書に代えて代理証明書を前記第１のサーバへ送信する第２のステップとを備える認証方法。

【請求項９】　前記第２のステップにおいて、前記代理証明書は、前記第２のサーバが証明書を保持するか否かに拘わらず前記第１のサーバへ送信される、請求項８に記載の認証方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、暗号通信システムに関し、特に、サーバからのクライアントの証明書の要求に対して代理応答する暗号通信システムおよびそれに用いる認証方法に関するものである。

【０００２】

【従来の技術】現在、インターネットを用いて各種の情報が通信されている。このインターネットを用いた通信においては、図１０に示すように、２つのコンピュータ１００，１１０は、ＯＳＩ（Ｏｐｅｎ　Ｓｙｓｔｅｍ　Ｉｎｔｅｒｃｏｎｎｅｃｔｉｏｎ）参照モデル１３０を用いて各種の情報を送受信する。コンピュータ１００は、ケーブル、無線等の物理的接続１２０によってコンピュータ１１０とデータ等の各種の情報をやり取りする。ＯＳＩ参照モデル１３０は、物理層、データリンク層、ネットワーク層、トランスポート層、セッション層、プレゼンテーション層、およびアプリケーション層の７層から成る。物理層が最下層であり、アプリケーション層が最上層である。また、第１層から第４層までが下位層であり、第５層から第７層が上位層である。コンピュータ１００がコンピュータ１１０と通信を行なうとき、各層は、それぞれ、機能の異なるプロトコルであるが、互いに同じ層のやり取りを意識することによって結果的に全体の通信プロトコルが成立する。

【０００３】図１１は、ＯＳＩ参照モデル１３０の各層の機能を示したものである。最上層のアプリケーション層は、コンピュータ１００，１１０を操作するユーザが使用するアプリケーション（サービス）と下の層との橋渡しを行なう。例えば、コンピュータ１００，１１０のユーザが通信すべきメッセージとして「こんにちは」とキーボードから入力したとき、これを下の層に引渡したり、逆に下の層からメッセージが届いた場合は、これをアプリケーションとして認識する働きをする。データ通

信用のアプリケーションには、ファイル転送や電子メールの送受信等、様々なものであるが、アプリケーション層では、届いたデータがこれらのアプリケーションのうち、どのアプリケーションのものかを判断し、受信側で該当する正しいアプリケーションに引渡す。また、送信される情報では、送信先のどのアプリケーションへ渡されるものかが明確になっている。従って、アプリケーション層は、これらの交通整理を的確に行ない、情報自体を認識し、制御する。

【０００４】プレゼンテーション層は、上の層から渡された情報を通信に適した形に変換して下の層に引渡したり、下の層から渡された情報をアプリケーション層に適した形に変換して上の層に引渡す。例えば、プレゼンテーション層は、上述した「こんにちは」というメッセージを符号して下の層に引渡し、下の層から渡された符号化されたデータを「こんにちは」というメッセージに変換して上の層へ渡す。セッション層は、プレゼンテーション層で符号化されたデータを相手に送信する機能を果たす。つまり、セッション層は、データを相手に送信する際、相手との通信経路を確立したり、通信方法を決定する。

【０００５】トランスポート層は、通信情報の質を高めるための通信制御を行なう。具体的には、トランスポート層は、相手との通信においてデータが欠落したとき、その欠落したデータを再送信してもらうように依頼する。ネットワーク層は、データを送信する際の送信元と送信先とを決定する。つまり、ネットワーク層は、データにヘッダを付加し、そのヘッダに送信元のアドレスと送信先のアドレスとを書込む。データリンク層は、ネットワーク層で設定された相手のアドレスに基づいて、データを次に送信すべき宛先を管理する。物理層は、データリンク層から渡されたビット情報を実際に伝送するための電気信号に変換したり、受信した電気信号をビット情報に変換する。

【０００６】上述した７層から成るＯＳＩ参照モデル１３０を用いて２つのコンピュータ１００，１１０間でデータが通信される。このようなインターネットを用いたデータ通信においては、データのセキュリティが重要な問題であり、データを暗号化して通信することが行なわれている。インターネットを用いた暗号通信で一般的に使用されているのは、ＳＳＬ（Ｓｅｃｕｒｅ　Ｓｏｃｋｅｔ　Ｌａｙｅｒ）暗号通信である。ＳＳＬ暗号通信は、公開鍵を基盤としたセキュリティ・インフラストラクチャーであるＰＫＩ（Ｐｕｂｌｉｃ　Ｋｅｙ　Ｉｎｆｒａｓｔｒｕｃｔｕｒｅ）の実装暗号化通信方式である。そして、ＳＳＬ暗号通信においては、送信データの暗号化に共通鍵暗号方式が用いられ、クライアントとサーバとが送信データの暗号化に用いる共通鍵を共有するために公開鍵暗号方式が用いられている。

【０００７】インターネットによる通信において、不正

なアクセスなどによる重要データの漏洩・破壊を防ぐためにファイアウォールを設け、通過するデータを制御することが行なわれている。しかし、ＳＳＬ暗号通信の場合、ＯＳＩ参照モデルの上層部が暗号化されているため、暗号化されていない平文での通信に比べファイアウォールで十分な通信制御を行なうことができない。このような問題を解決するために、”情報処理学会第５８回（平成１１年前期）全国大会５Ｎ－６，３－３３３～３－３３４”には、インターネットとの暗号データとのやり取りをサーバに代わって行なうＳＳＬ代理応答システムが開示されている。図１２は、このＳＳＬ代理応答システムを示したものである。ＳＳＬ代理応答システム２００は、インターネット２１０と、ファイアウォール２２０と、代理サーバ２３０と、サーバ２４０とから成る。ファイアウォール２２０は、インターネット２１０とサーバ２４０との間に配置され、代理サーバ２３０はファイアウォール２２０に接続されている。インターネット２１０からサーバ２４０へ暗号データが送信されるとき、ファイアウォール２２０は暗号データを代理サーバ２３０へ送信する。そして、代理サーバ２３０は、受信した暗号データを復号し、その復号した平文のデータをファイアウォール２２０へ送信する。ファイアウォール２２０は、代理サーバ２３０からの平文のデータに対して所定の制御を行なった後、サーバ２４０へ送信する。

【０００８】サーバ２４０からデータを送信するとき、サーバ２４０は、平文のデータをファイアウォール２２０へ送信する。ファイアウォール２２０は、受信したデータに対して所定の制御を行なった後、データを代理サーバ２３０へ送信する。代理サーバ２３０は、受信したデータを暗号化して暗号データをファイアウォール２２０へ送信する。ファイアウォール２２０は、受信した暗号データをインターネット２１０を介して送信する。

【０００９】このＳＳＬ代理応答システム２００においては、代理サーバ２３０がサーバ２４０に代わってデータの暗号化および復号化を行なうことにより、ファイアウォール２２０によるアプリケーション層の通信制御を可能にしている。また、暗号データの通信においては、通信相手が正規の相手であることを相互に認証することがセキュリティの面から重要であるが、ＳＳＬ代理応答システム２００においては、代理サーバ２３０は、証明書および秘密鍵をサーバ２４０から譲り受け、サーバ２４０に代わって暗号認証通信を行なう。

【００１０】また、暗号データの通信において相互認証するとき、その相互認証に用いる証明書が漏洩したとき、つまり、サーバの公開鍵やユーザの個人情報を暗号化する秘密鍵（この秘密鍵は認証機関によって認証されている）が漏洩した場合、不正に侵入してきた相手と暗号データを通信することになり、重要なデータが外部に漏洩する。このような問題を解決するために、”コンピ

ュータセキュリティ　７－４（２０００．１．２１）ｐ１９～２４”には、認証辞書（Ａｕｔｈｅｎｔｉｃａｔｅｄ　Ｄｉｃｔｉｏｎａｒｙ）を用いて通信相手から送信されてきた証明書が廃棄された証明書でないことを確認するシステムが開示されている。この認証辞書は、証明書が漏洩したことを示す証明書廃棄リスト（ＣＲＬ：Ｃｅｒｔｉｆｉｃａｔｅ　Ｒｅｖｏｃａｔｉｏｎ　Ｌｉｓｔ）と等価な証明書の廃棄情報を持つものである。このシステムにおいては、ＳＳＬ暗号通信における通信路の確立時にサーバから送信された証明書の廃棄情報をクライアントが確認する。そして、クライアントは、受信した証明書が廃棄情報に含まれていなければ、その証明書に基づいて通信相手を認証する。

【００１１】

【発明が解決しようとする課題】しかし、上述した従来の暗号データの通信方式においては、通信相手の認証時、暗号データの通信を行なう送信者と受信者との間で、自己の証明書を提示し合って認証するものである。そして、証明書には、ユーザの個人情報も含まれるため、通信相手の認証時に個人情報が相手に知られるという問題がある。図１２に示すＳＳＬ代理応答システム２００においても、代理サーバ２３０が暗号認証通信を行なうが、その際に送信するのはサーバ２４０の証明書である。したがって、サーバ２４０の所有者の個人情報が通信相手に知られる。

【００１２】また、従来の暗号データの通信方式においては、通信相手の相互認証は証明書に基づいており、いずれか一方の秘密鍵が漏洩した場合、第三者のなりすましが危惧される。しかし、現在のＳＳＬ暗号通信の確立段階で、証明書の有効性を検証する証明書廃棄リスト（ＣＲＬ）との照合が行なわれていないため、第三者のなりすましによる不正な暗号通信を防止できないという問題がある。

【００１３】そこで、本発明は、かかる問題を解決するためになされたものであり、その目的は、クライアントがサーバに対して自己の証明書を送信しなくても、相互認証が可能な暗号通信システムおよびそれに用いる認証方法を提供することである。

【００１４】また、本発明の別の目的は、さらに、漏洩した証明書を送信した相手との通信を防止できる暗号通信システムを提供することである。

【００１５】

【課題を解決するための手段】この発明による暗号通信システムは、複数の層構造から成るプロトコルを用いて、所定の暗号方式によって暗号化された暗号データの通信を行なう暗号通信システムであって、データまたは暗号データを送受信する第１のサーバと、第１のサーバとの間でデータまたは暗号データを送受信する第２のサーバと、第１のサーバと第２のサーバとの通信を中継する第３のサーバとを備え、第２のサーバの認証時、第３

のサーバは、第１のサーバからの第２のサーバの証明書の要求に応じて、第２のサーバが正規のサーバであることを示す代理証明書を第２のサーバに代わって第１のサーバへ送信する。

【００１６】この発明による暗号通信システムにおいては、第１のサーバが第２のサーバに対して証明書の送信を要求したとき、第３のサーバは、第２のサーバに代わって代理証明書を第１のサーバへ送信する。

【００１７】したがって、この発明によれば、第２のサーバは、個人情報を第１のサーバへ送信しなくても第１のサーバとの間で暗号通信を行なうことができる。

【００１８】好ましくは、暗号通信システムにおいて、第３のサーバは、自己の証明書を代理証明書として第１のサーバへ送信する。

【００１９】第３のサーバは、第２のサーバの証明書の代わりに自己の証明書を第１のサーバへ送信する。そして、第１のサーバは、代理証明書に基づいて第２のサーバを正規のサーバとして認証する。

【００２０】したがって、この発明によれば、第２のサーバは、自己の証明書を第１のサーバへ送信しなくても正規のサーバとして第１のサーバとの間で暗号通信を行なうことができる。

【００２１】好ましくは、暗号通信システムにおいて、第３のサーバは、第１のサーバから受信した第２のサーバの証明書の要求を第２のサーバへ送信し、第２のサーバから第２のサーバの証明書を受信すると代理証明書を第１のサーバへ送信する。

【００２２】第２のサーバは、第１のサーバからの第２のサーバの証明書の送信要求を受取り、自己の証明書を第３のサーバへ送信する。そうすると、第３のサーバは、第２のサーバから受信した証明書に代えて代理証明書を第１のサーバへ送信する。

【００２３】したがって、この発明によれば、第２のサーバが証明書を保持している場合でも第２のサーバの個人情報を第１のサーバに公表せずに第１のサーバと第２のサーバとの間で暗号通信を行なうことができる。

【００２４】好ましくは、暗号通信システムにおいて、第３のサーバは、第２のサーバの証明書の要求を第２のサーバへ送信し、第２のサーバが証明書を保持しないことを示す情報を第２のサーバから受信すると代理証明書を第１のサーバへ送信する。

【００２５】第２のサーバは、第１のサーバからの第２のサーバの証明書の送信要求を受取り、証明書を保持していないことを第３のサーバへ送信する。そうすると、第３のサーバは、自己が保持する代理証明書を第１のサーバへ送信する。

【００２６】したがって、この発明によれば、第２のサーバが証明書を保持していない場合でも、正規のサーバとして第１のサーバとの間で暗号通信を行なうことができる。

【００２７】好ましくは、暗号通信システムにおいて、第３のサーバは、第１のサーバから受信した第１のサーバの証明書を証明書廃棄リストと照合し、第１のサーバの証明書が廃棄されているとき第１のサーバの証明書が無効であることを示す情報を第１のサーバへ送信する。

【００２８】第１のサーバの証明書が証明書廃棄リストに含まれるとき、第１のサーバへ証明書の無効通知が送信される。

【００２９】したがって、この発明によれば、不正な相手との暗号通信を防止できる。好ましくは、暗号通信システムにおいて、第３のサーバは、第１のサーバから受信した第１のサーバの証明書を証明書廃棄リストと照合し、第１のサーバの証明書が廃棄されていないことを確認すると代理証明書を第１のサーバへ送信する。

【００３０】第２のサーバの証明書の送信を要求した第１のサーバが正規であることが確認されると、第３のサーバは、代理証明書を第１のサーバへ送信する。

【００３１】したがって、この発明によれば、暗号通信におけるセキュリティを向上させることができる。

【００３２】好ましくは、第３のサーバは、第１のサーバと第２のサーバとの間の通信を制御する通信制御部と、通信制御部を介して受取った第１のサーバの証明書を証明書廃棄リストと照合する証明書照合部と、証明書照合部からの照合結果に基づいて、第１のサーバを認証し、または第１のサーバの証明書を無効と判定する認証部と、通信制御部を介して第２のサーバの証明書または第２のサーバが証明書を保持しない情報を受取ると、代理証明書を第１のサーバへ送信する代理応答部とを含む。

【００３３】第３のサーバは、第１のサーバと第２のサーバとの間の通信を制御するとともに、証明書の証明書廃棄リストとの照合、証明書の認証、および代理応答を行なう。

【００３４】したがって、この発明によれば、暗号通信を行なう２つのサーバの間に、代理応答等を行なうサーバを設けることによって個人情報を隠匿して暗号通信を行なうことができる。

【００３５】また、この発明による認証方法は、第１のサーバと第２のサーバとの間における認証方法であって、第２のサーバの証明書の送信要求を第１のサーバから受信する第１のステップと、第２のサーバの証明書に代えて代理証明書を第１のサーバへ送信する第２のステップとを備える。

【００３６】この発明による認証方法においては、第１のサーバからの証明書の送信要求に対して代理証明書が第１のサーバへ送信されて第２のサーバの認証が行なわれる。

【００３７】したがって、この発明によれば、個人情報を相手に公開せずとも相互認証を行なうことができる。

【００３８】好ましくは、第２のステップにおいて、代理証明書は、第２のサーバが証明書を保持するか否かに拘わらず第１のサーバへ送信される。

【００３９】第２のサーバが認証機関によって認証された証明書を保持しているか否かに関係なく代理証明書が第１のサーバへ送信され、第２のサーバの認証が行なわれる。

【００４０】したがって、この発明によれば、証明書を保持しないサーバでも正規のサーバとして暗号通信を行なうことができる。

【００４１】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照しながら詳細に説明する。なお、図中同一または相当部分には同一符号を付してその説明は繰返さない。

【００４２】図１は、本発明による暗号通信システムの概略ブロック図である。暗号通信システム１０は、サーバ１と、クライアントサーバ２Ａ，２Ｂ，２Ｃと、インターネット網３と、ＣＡ　Ｐｒｏｘｙ（Ｃｅｒｔｉｆｉｃａｔｅ　Ａｕｔｈｏｒｉｔｙ　Ｐｒｏｘｙ）４と、結合器５とを備える。サーバ１は、インターネット網３に接続される。クライアントサーバ２Ａ，２Ｂ，２Ｃは、結合器５に接続される。ＣＡ　Ｐｒｏｘｙ４は、インターネット網３と結合器５との間に配置される。

【００４３】サーバ１は、後述する方法によってインターネット網３、ＣＡ　Ｐｒｏｘｙ４、および結合器５を介してクライアントサーバ２Ａ，２Ｂ，２Ｃへデータまたは暗号データを送信し、またはクライアントサーバ２Ａ，２Ｂ，２Ｃからデータまたは暗号データを受信する。インターネット網３は、サーバ１からのデータまたは暗号データをＣＡ　Ｐｒｏｘｙ４へ送信し、ＣＡ　Ｐｒｏｘｙ４からのデータまたは暗号データをサーバ１へ送信する。

【００４４】ＣＡ　Ｐｒｏｘｙ４は、インターネット網３からのデータまたは暗号データを結合器５を介してクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する。また、ＣＡＰｒｏｘｙ４は、クライアントサーバ２Ａ，２Ｂ，２Ｃからのデータまたは暗号データを結合器５を介して受信し、その受信したデータまたは暗号データをインターネット網３を介してサーバ１へ送信する。さらに、ＣＡ　Ｐｒｏｘｙ４は、サーバ１から受信した電子証明書を証明書廃棄リスト（ＣＲＬ）と照合し、受信した電子証明書がＣＲＬに含まれていないとき、受信した電子証明書に基づいてサーバ１を正規の通信相手として認証する。また、さらに、ＣＡ　Ｐｒｏｘｙ４は、サーバ１からクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書の提出要求に応じて、自己の電子証明書をサーバ１へ送信する。即ち、ＣＡ　Ｐｒｏｘｙ４は、クライアントサーバ２Ａ，２Ｂ，２Ｃが電子証明書を保持しているか否かに拘わらず、サーバ１に対してクライアントサーバ２Ａ，２Ｂ，２Ｃが正規のサーバであることを示す自己の

電子証明書を送信する。つまり、ＣＡ　Ｐｒｏｘｙ４は、クライアントサーバ２Ａ，２Ｂ，２Ｃに代わってサーバ１に対して電子証明書の代理応答を行なう。これにより、クライアントサーバ２Ａ，２Ｂ，２Ｃは、自己の電子証明書をサーバ１へ送信しなくても、相互認証ができ、サーバ１との間で通信路を確立できる。

【００４５】結合器５は、ＣＡ　Ｐｒｏｘｙ４からのデータまたは暗号データをクライアントサーバ２Ａ，２Ｂ，２Ｃの各々へ振り分ける。

【００４６】図２は、サーバ１、クライアントサーバ２Ａ，２Ｂ，２Ｃ、およびＣＡ　Ｐｒｏｘｙ４の機能ブロックを示したものである。サーバ１は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１と、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２と、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３と、ＡｌｅｒｔＰｒｏｔｏｃｏｌ部１４と、Ａｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部１５と、アプリケーション部１６とを含む。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３、Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部１４、およびＡｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｔｏｃｏｌ部１５はＳＳＬプロトコルを構成する。ＳＳＬプロトコルは、図１０に示したセッション層のプロトコルであり、アプリケーション部１６は、図１０に示したセッション層よりも上位のプロトコルである。

【００４７】Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、アプリケーション部１６からのデータを圧縮／暗号化して、クライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する。また、クライアント２Ａ，２Ｂ，２Ｃから受信した暗号データを復号化／伸張してアプリケーション部１６へ引渡す。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、暗号化アルゴリズム、暗号鍵、電子証明書など、暗号データの通信を開始するために必要なパラメータを通信相手と交渉し、決定する。

【００４８】Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３は、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２で決定された、新しい暗号化通信パラメータの利用開始を通信相手に通知し、自らも利用を開始する。Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部１４は、通信中に発生したイベントやエラーを通信相手に通知する。Ａｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部１５は、現在、有効な暗号化通信パラメータに従って、アプリケーションデータを透過的に送受信する。アプリケーション部１６は、ＳＳＬプロトコルのＡｐｐｌｉｃａｔｉｏｎＤａｔａ　Ｐｒｏｔｏｃｏｌ部１５からのデータを受取って、その受取ったデータを処理する。また、アプリケーション部１６は、新たに作成したデータをＳＳＬプロトコルのＡｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部１５へ引渡す。

【００４９】クライアントサーバ２Ａ，２Ｂ，２Ｃは、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１と、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２と、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３と、Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部２４と、Ａｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部２５と、アプリケーション部２６とを含む。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３、Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部２４、およびＡｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部２５は、ＳＳＬプロトコルを構成する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３、Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部２４、およびＡｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部２５は、それぞれ、サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３、Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部１４、およびＡｐｐｌｉｃａｔｉｏｎ　ＤａｔａＰｒｏｔｏｃｏｌ部１５と同じ機能を果たす。また、アプリケーション部２６は、サーバ１のアプリケーション部１６に対応する。

【００５０】ＣＡ　Ｐｒｏｘｙ４は、通信プロトコルキャプチャー部４１と、ＣＲＬチェック部４２と、認証部４３と、代理応答部４４とを含む。通信プロトコルキャプチャー部４１は、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間で行なわれる通信を監視するとともに、ＣＲＬチェック部４２、認証部４３、および代理応答部４４で処理すべき通信プロトコルをそれぞれに振り分ける。ＣＲＬチェック部４２は、第三者である認証機関（図示せず）が作成した証明書廃棄リスト（ＣＲＬ）を認証機関から取得して保持しており、通信プロトコルキャプチャー部４１がサーバ１から受信したサーバ１の電子証明書を受け、サーバ１の電子証明書を証明書廃棄リストと照合し、その照合結果を通信プロトコルキャプチャー部４１へ出力する。なお、ＣＲＬチェック部４２は、認証機関から証明書廃棄リストを定期的に取得し、証明書廃棄リストの更新を行なう。

【００５１】認証部４３は、ＣＡ　Ｐｒｏｘｙ４の電子証明書を保持しており、サーバ１からクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書の提出を要求されると、クライアントサーバ２Ａ，２Ｂ，２ＣはＣＡ　Ｐｒｏｘｙ４の支配下にあるクライアントサーバであることの認証を行なう。この場合、認証部４３は、クライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書に代えてＣＡ　Ｐｒｏｘｙ４の電子証明書を代理応答部４４へ出力する。代理応答部４４は、クライアントサーバ２Ａ，２

Ｂ，２Ｃから”Ｃｌｉｅｎｔ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージまたは”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを通信プロトコルキャプチャー部４１を介して受信すると、認証部４３から受取ったＣＡ　Ｐｒｏｘｙ４の電子証明書をクライアントサーバ２Ａ，２Ｂ，２Ｃに代わってサーバ１へ送信する。

【００５２】なお、サーバ１およびクライアントサーバ２Ａ，２Ｂ，２ＣにおけるＳＳＬプロトコルはＩＥＴＦ（Ｉｎｔｅｒｎａｔｉｏｎａｌ　ＥｎｇｉｎｅｅｒｉｎｇＴａｓｋ　Ｆｏｒｃｅ）によって公開されている機能と同じである。

【００５３】図３は、電子証明書の構成を示す概略ブロック図である。電子証明書５０は、バージョン５１と、シリアル番号５２と、証明書発行者５３と、発行者ユニーク識別子５４と、証明対象ユーザ５５と、ユーザユニーク識別子５６と、証明対象公開鍵アルゴリズム５７と、証明対象公開鍵５８と、証明書有効期限５９と、証明書拡張６０と、署名アルゴリズム６１と、署名６２とを含む。バージョン５１は、証明書５０の構成要素を規定するものである。シリアル番号５２は、その証明書が何番目に発行されたかを示す番号である。発行者ユニーク識別子５４は、証明書を発行する認証機関を識別するための情報であり、認証機関に固有のＩＤが書込まれる。証明対象ユーザ５５は、認証機関に自己の公開鍵の認証を依頼するユーザの名前である。ユーザユニーク識別子５６は、認証機関に自己の公開鍵の認証を依頼するユーザを識別するための情報であり、ユーザＩＤが書込まれる。証明対象公開鍵アルゴリズム５７は、ユーザが認証を依頼した公開鍵を生成するためのアルゴリズムである。証明対象公開鍵５８は、ユーザに依頼されて、認証機関が認証する公開鍵である。証明書有効期限５９は、電子証明書５０が有効である期間である。証明書拡張６０は、将来、各種の情報を電子証明書に格納するための枠組みを与えるものである。署名アルゴリズム６１は、認証機関が署名するとき、つまり、電子証明書５０のバージョン５１から署名アルゴリズム６１の各情報を認証機関の秘密鍵で暗号化し、または認証機関の復号鍵で復号するときのアルゴリズムである。署名６２は、認証機関が電子証明書５０のバージョン５１から署名アルゴリズム６１の各情報を自己の秘密鍵で暗号化していることを示す情報である。暗号通信システム１０は、データを暗号化する共通鍵をサーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間で共有するために公開鍵暗号方式を用いている。従って、サーバ１、およびクライアントサーバ２Ａ，２Ｂ，２Ｃは、自己の公開鍵５８を第三者である認証機関へ登録し、その登録によって自己の公開鍵５８を認証してもらう。そして、サーバ１、およびクライアントサーバ２Ａ，２Ｂ，２Ｃは、自己の公開鍵５８を認証機関で認証してもらった電子証明書５０を保持し、その電子証明書５０によって自己が認証機関によって認証された正規のサーバであることを証明する。証明対象ユーザ５５およびユーザユニーク識別子５６は、ユーザの個人情報に該当する部分である。

【００５４】従って、電子証明書５０を受信したサーバ１等は、電子証明書５０を認証機関が発行した公開鍵によって復号すれば、電子証明書５０を送信した相手が正規のサーバであるか否かを判別できる。

【００５５】図４～図６は、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間のＳＳＬ暗号通信におけるセッション確立のフローチャートである。まず、図４に示すフローチャートについて説明する。サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間の通信が開始されると（ステップＳ１００）、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介して送信する（ステップＳ１０２）。この”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージは、通信プロトコルのバージョン、セッションＩＤ、暗号化アルゴリズム等の候補を含む。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、クライアントサーバ２Ａ，２Ｂ，２Ｃからの”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージを受信し、その受信した”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージをサーバ１へ送信する。サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、ＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部１１を介して”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージを受信する（ステップＳ１０４）。そして、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、受信した”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージに含まれるプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムの候補から１つのプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを選択し、その選択したプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを”ＳｅｖｅｒＨｅｌｌｏ”メッセージに含めてクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ１０６）。

【００５６】ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１からの”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージを受信し、その受信した”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージをクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する。クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージをＲｅｃｏｒ　Ｐｒｏｔｏｃｏｌ部２１を介して受信し、”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージに基づいてサーバ１が選択したプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを確認する（ステップＳ１０８）。これによって、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間の暗号通信方式が決定される。

【００５７】その後、サーバ１のＨａｎｄｓｈａｋｅ

Ｐｒｏｔｏｃｏｌ部１２は、”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して送信する（ステップＳ１１０）。なお、この”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”は、サーバ１の電子証明書であり、通信関連の国際標準機構であるＩＴＵ（Ｉｎｔｅｒｎａｔｉｏｎａｌ　Ｔｅｌｅｃｏｍｍｕｎｉｃａｔｉｏｎ　Ｕｎｉｏｎ）で標準化されたＸ．５０９ｖ３による標準仕様に従って作成されている。以下に述べる”ＣｌｉｅｎｔＣｅｒｔｉｆｉｃａｔｅ”も同様である。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１からの”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”を受信し、その受信した”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージをＣＲＬチェック部４２へ出力する。ＣＲＬチェック部４２は、”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージを受取り、サーバ１の電子証明書を証明書廃棄リスト（ＣＲＬ）と照合し、サーバ１の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれるか否かをチェックする。そして、ＣＲＬチェック部４２は、照合結果を通信プロトコルキャプチャー部４１へ出力する（ステップＳ１１２）。

【００５８】サーバ１の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれる場合、通信プロトコルキャプチャー部４１は、サーバ１の電子証明書が無効であることを示す無効通知をサーバ１へ送信する（ステップＳ１１４）。そして、サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して無効通知を受信し（ステップＳ１１６）、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの通信は終了する（ステップＳ１５４）。つまり、サーバ１は、正規のサーバではないと判断されたので、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの通信は終了し、クライアントサーバ２Ａ，２Ｂ，２Ｃの重要な情報が不正なサーバへ漏洩するのを防止できる。

【００５９】ステップＳ１１２において、サーバ１の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれていないと判断されたとき、ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１から受信した電子証明書をクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信し（ステップＳ１１８）、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介してサーバ１の電子証明書を受信する（ステップＳ１２０）。これによって、サーバ１は正規のサーバであることが認証されるとともに、クライアントサーバ２Ａ，２Ｂ，２Ｃは、サーバ１の公開鍵を取得する。

【００６０】そして、サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、クライアントサーバ２Ａ，２Ｂ，２Ｃに対して電子証明書の送信を要求するか否かを判定する（ステップＳ１２２）。電子証明書の送信を要求しないとき、図５に示すステップＳ１３２へ移行する。また、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、電子証明書の提出を要求すると判定したとき、”Ｃｅｒｔｉｆｉｃａｔｅ　Ｒｅｑｕｅｓｔ”をＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して送信し、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、ＣＡ　Ｐｒｏｘｙ４およびＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介して”Ｃｅｒｔｉｆｉｃａｔｅ　Ｒｅｑｕｅｓｔ”を受信し、それに対して”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージをＣＡ　Ｐｒｐｘｙ４へ送信する（ステップＳ１２４）。

【００６１】次に、図５に示すフローチャートについて説明する。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、クライアントサーバ２Ａ，２Ｂ，２Ｃから”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを受信し、その受信した”ＮｏＣｅｒｔｉｆｉｃａｔｅ”メッセージを認証部４３および代理応答部４４へ出力する。そして、認証部４３は、”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを受取ると、保持しているＣＡ　Ｐｒｏｘｙ４の電子証明書を代理応答部４４へ出力する。代理応答部４４は、通信プロトコルキャプチャー部４１からの”ＮｏＣｅｒｔｉｆｉｃａｔｅ”メッセージを受けると、サーバ１に対して代理応答するか否かを判定する（ステップＳ１２６）。代理応答部４４が代理応答しないと判定したとき、ステップＳ１５４へ移行し、通信は終了する。代理応答部４４は、代理応答すると判定すると、認証部４３から入力されたＣＡ　Ｐｒｐｘｙ４の電子証明書を通信プロトコルキャプチャー部４１を介してサーバ１へ送信する（ステップＳ１２８）。

【００６２】サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、ＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部１１を介してＣＡ　Ｐｒｏｘｙ４からの電子証明書を受信し、その受信した電子証明書に基づいてＣＡ　Ｐｒｏｘｙ４が正規のサーバであることを認証するとともに、クライアントサーバ２Ａ，２Ｂ，２Ｃとの暗号通信に用いる公開鍵を取得する（ステップＳ１３０）。この場合、ＨａｎｄｓｈａｋｅＰｒｏｔｏｃｏｌ部１２は、形式的にはＣＡ　Ｐｒｏｘｙ４を正規のサーバとして認証するが、ＣＡ　Ｐｒｏｘｙ４はクライアントサーバ２Ａ，２Ｂ，２Ｃに代わって電子証明書をサーバ１へ送信しているので、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、実質的にはクライアントサーバ２Ａ，２Ｂ，２Ｃを正規のサーバとして認証する。また、ＣＡ　Ｐｒｏｘｙ４は、クライアントサーバ２Ａ，２Ｂ，２Ｃが電子証明書を保持しない場合でもクライアントサーバ２Ａ，２Ｂ，２Ｃに代わって自己の電子証明書をサーバ１へ送信するので、認証機関によって認証された電子証明書を保持しないクライアントでも正規のサーバとしてサーバ１との暗号通信が可能になる。

【００６３】ステップＳ１２２で”Ｎｏ”が選択された後、またはステップＳ１３０の後、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、４８バイトの乱数を発生させ、その発生させた乱数をＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、入力された乱数をサーバ１の公開鍵Ｐａで暗号化し、その暗号化した乱数を”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージとしてサーバ１へ送信する（ステップＳ１３２）。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２が発生した乱数は、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間でデータを暗号通信する際の共通鍵を生成するためのものであり、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、発生した乱数を用いて共通鍵を生成する。

【００６４】一方、サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、ＣＡ　Ｐｒｏｃｙ４を介して”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージを受信し、暗号化された乱数を秘密鍵Ｓａで復号する（ステップＳ１３４）。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号した乱数をＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２へ出力する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、入力された４８バイトの乱数を用いて共通鍵を生成する。

【００６５】その後、クライアントサーバ２Ａ，２Ｂ，２ＣのＣｈａｎｇｅ　ＣｉｐｈｅｒＳｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３は、ステップＳ１３６までにサーバ１とクライアント２Ａ，２Ｂ，２Ｃとの間で合意された暗号通信方式に同意し、生成した共通鍵を認めることを示す”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを生成してＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。ＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部２１は、共通鍵ＫＰａによって”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを暗号化した｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを生成してクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ１３６）。

【００６６】サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、ＣＡ　Ｐｒｏｘｙ４を介して｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを受信し、共通鍵ＫＰａを用いて｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを復号する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号した”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージをＣｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ部１３へ出力する。Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３は、”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを受けて、クライアントサーバ２Ａ，２Ｂ，２Ｃが暗号通信方式や共通鍵に同意したことを検知する（ステップＳ１３８）。そして、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅＰｒｏｔｏｃｏｌ部２２は、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌの終了を表す”Ｆｉｎｉｓｈｅｄ”メッセージを生成してＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、上記で決めた暗号化仕様に従って共通鍵ＫＰａで”Ｆｉｎｉｓｈｅｄ”メッセージを暗号化し、｛Ｆｉｎｉｓｈｅｄ｝ＫＰａをサーバ１へ出力する（ステップＳ１４０）。

【００６７】サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、ＣＡ　Ｐｒｏｘｙ４を介して｛Ｆｉｎｉｓｈｅｄ｝ＫＰａを受信し、その受信した｛Ｆｉｎｉｓｈｅｄ｝ＫＰａを共通鍵ＫＰａによって復号する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号した”Ｆｉｎｉｓｈｅｄ”メッセージをＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２へ出力する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、”Ｆｉｎｉｓｈｅｄ”メッセージを受理する（ステップＳ１４２）。

【００６８】最後に、図６に示すフローチャートについて説明する。その後、Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部１３は、”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを生成してＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部１１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、共通鍵ＫＰａによって”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを暗号化し、その暗号化した｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａをクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ１４４）。

【００６９】クライアントサーバ２Ａ，２Ｂ，２ＣのＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、ＣＡ　Ｐｒｏｘｙ４を介して｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを受信し、共通鍵ＫＰａによって｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを復号する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、復号した”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージをＣｈａｎｇｅ　ＣｉｐｈｅｒＳｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３へ出力する。Ｃｈａｎｇｅ　ＣｉｐｈｅｒＳｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３は、”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを受けてサーバ１が暗号通信方式や共通鍵に同意したことを検知する（ステップＳ１４６）。

【００７０】その後、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、クライアントサーバ２Ａ，２Ｂ，２Ｃへ送信するＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌの終了を示す”Ｆｉｎｉｓｈｅｄ”メッセージを生成してＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、上記で決めた暗号化仕様に従って共通鍵ＫＰａによって”Ｆｉｎｉｓｈｅｄ”メッセージを暗号化し、｛Ｆｉｎｉｓｈｅｄ｝ＫＰａをクライアントサーバ２Ａ，２Ｂ，２Ｃへ出力する（ステップＳ１４８）。クライアントサーバ２Ａ，２Ｂ，２ＣのＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、

ＣＡ　Ｐｒｏｘｙ４を介して｛Ｆｉｎｉｓｈｅｄ｝ＫＰａを受信し、その受信した｛Ｆｉｎｉｓｈｅｄ｝ＫＰａを共通鍵ＫＰａによって復号する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、復号した”Ｆｉｎｉｓｈｅｄ”メッセージをＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２へ出力する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”Ｆｉｎｉｓｈｅｄ”メッセージを受理する（ステップＳ１５０）。

【００７１】ステップＳ１５０までででサーバ１とクライアントサーバ２Ａ，２Ｂ，２ＣとのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ、即ち、セッションの確立が終了する。そして、サーバ１のＡｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部１５とアプリケーション部１６、およびクライアントサーバ２Ａ，２Ｂ，２ＣのＡｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部２５とアプリケーション部２６による共通鍵ＫＰａを用いた暗号通信が行なわれて（ステップＳ１５２）、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間の通信が終了する（ステップＳ１５４）。

【００７２】図４から図６に示したフローチャートは、クライアントサーバ２Ａ，２Ｂ，２Ｃが自己の電子証明書を保持しない場合のサーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとのセッション確立時のフローチャートである。上述したように、クライアントサーバ２Ａ，２Ｂ，２Ｃが自己の電子証明書を保持しない場合でも、ＣＡ　Ｐｒｏｘｙ４は、クライアントサーバ２Ａ，２Ｂ，２Ｃが正規のサーバであることを示すＣＡ　Ｐｒｏｘｙ４の電子証明書をサーバ１へ代理応答し、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間で相互認証が行なわれる。

【００７３】図７～図９は、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとのセッション確立時の別のフローチャートである。まず、図７に示すフローチャートについて説明する。サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間の通信が開始されると（ステップＳ２００）、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介して送信する（ステップＳ２０２）。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、クライアントサーバ２Ａ，２Ｂ，２Ｃからの”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージを受信し、その受信した”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージをサーバ１へ送信する。サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージを受信する（ステップＳ２０４）。そして、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、受信した”ＣｌｉｅｎｔＨｅｌｌｏ”メッセージに含まれるプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムの候補から１つのプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを選択し、その選択したプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを”ＳｅｖｅｒＨｅｌｌｏ”メッセージに含めてクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ２０６）。

【００７４】ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１からの”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージを受信し、その受信した”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージをクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する。クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージをＲｅｃｏｒ　Ｐｒｏｔｏｃｏｌ部２１を介して受信し、”ＳｅｒｖｅｒＨｅｌｌｏ”メッセージに基づいてサーバ１が選択したプロトコルバージョン、セッションＩＤ、および暗号アルゴリズムを確認する（ステップＳ２０８）。これによって、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間の暗号通信方式が決定される。

【００７５】その後、サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、電子証明書を保持するか否かを判定し（ステップＳ２１０）、電子証明書を保持していないと判定したとき、ステップＳ２２４へ移行する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、電子証明書を保持していると判定したとき、”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して送信する（ステップＳ２１２）。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１からの”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”を受信し、その受信した”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージをＣＲＬチェック部４２へ出力する。ＣＲＬチェック部４２は、”ＳｅｒｖｅｒＣｅｒｔｉｆｉｃａｔｅ”メッセージを受取り、サーバ１の電子証明書を証明書廃棄リスト（ＣＲＬ）と照合し、サーバ１の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれるか否かをチェックする。そして、ＣＲＬチェック部４２は、照合結果を通信プロトコルキャプチャー部４１へ出力する（ステップＳ２１４）。

【００７６】サーバ１の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれる場合、通信プロトコルキャプチャー部４１は、サーバ１の電子証明書が無効であることを示す無効通知をサーバ１へ送信する（ステップＳ２１６）。そして、サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１を介して無効通知を受信し（ステップＳ２１８）、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの通信は終了する（ステップＳ２７６）。つまり、サーバ１は、正規のサーバではないと判断されたので、サーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの通信

は終了し、クライアントサーバ２Ａ，２Ｂ，２Ｃの重要な情報が不正なサーバへ漏洩するのを防止できる。

【００７７】ステップＳ２１４において、サーバ1の電子証明書が証明書廃棄リスト（ＣＲＬ）に含まれていないと判断されたとき、ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、サーバ１から受信した電子証明書をクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信し（ステップＳ２２０）、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介してサーバ１の電子証明書を受信する（ステップＳ２２２）。これによって、サーバ１は正規のサーバであることが認証されるとともに、クライアントサーバ２Ａ，２Ｂ，２Ｃは、サーバ１の公開鍵を取得する。

【００７８】一方、ステップＳ２１０においては、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２が電子証明書を保持していないと判定したとき、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、”ＳｅｒｖｅｒＫｅｙＥｘｃｈａｎｇｅ”メッセージを生成してＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１へ出力する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、”ＳｅｒｖｅｒＫｅｙＥｘｃｈａｎｇｅ”メッセージをクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ２２４）。この”ＳｅｒｖｅｒＫｅｙＥｘｃｈａｎｇｅ”メッセージは、ＲＳＡ公開鍵またはＤｉｆｆｉｅ＆Ｈｅｌｌｍａｎ公開情報から成る。サーバ１が電子証明書を保持しない場合に、ＲＳＡ公開鍵またはＤｉｆｆｉｅ＆Ｈｅｌｌｍａｎ公開情報をクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信することによってサーバ１が正規のサーバであることを電子証明書に代えて証明するものである。

【００７９】ステップＳ２２２またはステップＳ２２４の後、サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、クライアントサーバ２Ａ，２Ｂ，２Ｃに対して電子証明書の送信を要求するか否かを判定する（ステップＳ２２６）。電子証明書の送信を要求しないとき、図９に示すステップＳ２５４へ移行する。また、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、電子証明書の提出を要求すると判定したとき、”Ｃｅｒｔｉｆｉｃａｔｅ　Ｒｅｑｕｅｓｔ”をＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部１１を介して送信し、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、ＣＡ　Ｐｒｏｘｙ４およびＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介して”Ｃｅｒｔｉｆｉｃａｔｅ　Ｒｅｑｕｅｓｔ”を受信する。

【００８０】次に、図８に示すフローチャートについて説明する。クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、電子証明書を保持しているか否かを判定する（ステップＳ２２８）。そして、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、電子証明書を保持していると判定したとき、”ＣｌｉｅｎｔＣｅｒｔｉｆｉｃａｔｅ”メッセージをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１を介してＣＡ　Ｐｒｏｘｙ４へ送信する（ステップＳ２３０）。ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、クライアントサーバ２Ａ，２Ｂ，２Ｃから”ＣｌｉｅｎｔＣｅｒｔｉｆｉｃａｔｅ”メッセージを受信し、その受信した”ＣｌｉｅｎｔＣｅｒｔｉｆｉｃａｔｅ”メッセージを認証部４３および代理応答部４４へ出力する。そして、認証部４３は、”ＣｌｉｅｎｔＣｅｒｔｉｆｉｃａｔｅ”メッセージを受取ると、そのメッセージに含まれている電子証明書に基づいてクライアントサーバ２Ａ，２Ｂ，２Ｃを正規のサーバとして認証し、保持しているＣＡ　Ｐｒｏｘｙ４の電子証明書を代理応答部４４へ出力する（ステップＳ２３２）。

【００８１】ステップＳ２２８において、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２が電子証明書を保持していないと判定したとき、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、”ＮｏＣｅｒｔｉｆｉｃａｔｅ”メッセージをＣＡ　Ｐｒｏｘｙ４へ送信する（ステップＳ２３４）。そして、ＣＡ　Ｐｒｏｘｙ４の通信プロトコルキャプチャー部４１は、クライアントサーバ２Ａ，２Ｂ，２Ｃから”ＮｏＣｅｒｔｉｆｉｃａｔｅ”メッセージを受信し、その受信した”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを認証部４３および代理応答部４４へ出力する。認証部４３は、”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを受取ると、保持しているＣＡ　Ｐｒｏｘｙ４の電子証明書を代理応答部４４へ出力する（ステップＳ２３６）。

【００８２】ステップＳ２３２またはステップＳ２３６の後、代理応答部４４は、通信プロトコルキャプチャー部４１からの”ＣｌｉｅｎｔＣｅｒｙｉｆｉｃａｔｅ”メッセージまたは”Ｎｏ　Ｃｅｒｔｉｆｉｃａｔｅ”メッセージを受けると、サーバ１に対して代理応答するか否かを判定する（ステップＳ２３８）。代理応答部４４が代理応答しないと判定したとき、ステップＳ２７６へ移行し、通信は終了する。代理応答部４４は、代理応答すると判定すると、認証部４３から入力されたＣＡ　Ｐｒｐｘｙ４の電子証明書を通信プロトコルキャプチャー部４１を介してサーバ１へ送信する（ステップＳ２４０）。

【００８３】サーバ１のＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、ＲｅｃｏｒｄＰｒｏｔｏｃｏｌ部１１を介してＣＡ　Ｐｒｏｘｙ４からの電子証明書を受信し、その受信した電子証明書に基づいてＣＡ　Ｐｒｏｘｙ４が正規のサーバであることを認証するとともに、クライアントサーバ２Ａ，２Ｂ，２Ｃとの暗号通信に用いる公開鍵を取得する（ステップＳ２４２）。この場合、ＨａｎｄｓｈａｋｅＰｒｏｔｏｃｏｌ部１２は、上述したように、クライアントサーバ２Ａ，２Ｂ，２Ｃを正規のサーバとして認証する。

【００８４】その後、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、４８バイトの乱数を発生させ、その発生させた乱数をＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、入力された乱数を公開鍵Ｐａで暗号化し、その暗号化した乱数を”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージとしてサーバ１へ送信する（ステップＳ２４４）。また、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、発生した乱数を用いてサーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間でデータを暗号通信する際の共通鍵を生成する。

【００８５】一方、サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、ＣＡ　Ｐｒｏｃｙ４を介して”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージを受信し、暗号化された乱数を秘密鍵Ｓａで復号する（ステップＳ２４６）。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号した乱数をＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２へ出力する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、入力された４８バイトの乱数を用いて共通鍵を生成する。

【００８６】クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、再度、電子証明書を保持するか否かを判定する（ステップＳ２４８）。電子証明書が保持されているときステップＳ２５０へ移行し、保持されていないときステップＳ２５８へ移行する。

【００８７】最後に、図９に示すフローチャートについて説明する。ステップＳ２４８において、電子証明書が保持されていると判定されたとき、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、サーバ１がクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書が正しいことを確認するために、ステップＳ２４８までに取得されメッセージのダイジェストをＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、入力されたメッセージのダイジェストを秘密鍵Ｓａで暗号化し、”ＣｅｒｔｉｆｉｃａｔｅＶｅｒｉｆｙ”としてサーバ１へ送信する（ステップＳ２５０）。サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｃｏｌ部１１は、”ＣｅｒｔｉｆｉｃａｔｅＶｅｒｉｆｙ”をＣＡ　Ｐｒｏｘｙ４を介して受信し、暗号化されたメッセージのダイジェストを公開鍵で復号する。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号したメッセージのダイジェストをＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２へ出力し、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、入力されたメッセージのダイジェストに基づいてクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書が正しいを確認する（ステップＳ２５２）。

【００８８】一方、ステップＳ２２６（図７参照）において、サーバ１がクライアントサーバ２Ａ，２Ｂ，２Ｃの電子証明書の送信を要求しないと判定としたとき、クライアントサーバ２Ａ，２Ｂ，２ＣのＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、４８バイトの乱数を発生させ、その発生させた乱数をＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、入力された乱数をサーバ１の公開鍵Ｐａで暗号化し、その暗号化した乱数を”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージとしてサーバ１へ送信する（ステップＳ２５４）。また、Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部２２は、発生した乱数を用いてサーバ１とクライアントサーバ２Ａ，２Ｂ，２Ｃとの間でデータを暗号通信する際の共通鍵を生成する。

【００８９】一方、サーバ１のＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、ＣＡ　Ｐｒｏｃｙ４を介して”ＣｌｉｅｎｔＫｅｙＥｘｃｈａｎｇｅ”メッセージを受信し（、暗号化された乱数を秘密鍵Ｓａで復号する（ステップＳ２５６）。そして、Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部１１は、復号した乱数をＨａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２へ出力する。Ｈａｎｄｓｈａｋｅ　Ｐｒｏｔｏｃｏｌ部１２は、入力された４８バイトの乱数を用いて共通鍵を生成する。

【００９０】ステップＳ２４８においてクライアントサーバ２Ａ，２Ｂ，２Ｃで電子証明書が保持されていないと判定されたとき、またはステップＳ２５２，Ｓ２５６の後、クライアントサーバ２Ａ，２Ｂ，２ＣのＣｈａｎｇｅ　Ｃｉｐｈｅｒ　Ｓｐｅｃ　Ｐｒｏｔｏｃｏｌ部２３は、ステップＳ２５６までにサーバ１とクライアント２Ａ，２Ｂ，２Ｃとの間で合意された暗号通信方式に同意し、生成した共通鍵を認めることを示す”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを生成してＲｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１へ出力する。Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部２１は、共通鍵ＫＰａによって”ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ”メッセージを暗号化した｛ＣｈａｎｇｅＣｉｐｈｅｒＳｐｅｃ｝ＫＰａを生成してクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信する（ステップＳ２５８）。

【００９１】それ以後のステップＳ２６０～Ｓ２７６は、図５および図６のステップＳ１３８～１５４と同じである。

【００９２】図７～図９に示すフローチャートは、サーバ１が電子証明書を保持しない場合に、ＲＳＡ公開鍵またはＤｉｆｆｉｅ＆Ｈｅｌｌｍａｎ公開情報をクライアントサーバ２Ａ，２Ｂ，２Ｃへ送信してサーバ１の正当性を認証してもらう点が、図４～図６に示したフローチャートと特に異なる点である。

【００９３】上記においては、サーバとクライアントサーバとの間の暗号通信方式をＳＳＬ暗号通信として説明したが、本発明は、これに限られるものではなく、公開鍵暗号方式であれば、どのような暗号方式を用いたものであっても良い。

23

【００９４】この発明の実施の形態によれば、暗号通信システムは、サーバに対してクライアントサーバが正規のサーバであることを代理応答するＣＡ　Ｐｒｏｘｙを備えるので、クライアントの個人情報がサーバに公開されることなく、サーバとクライアントサーバとの間で相互に認証し、暗号通信を行なうことができる。

【００９５】今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【００９６】

【発明の効果】この発明による暗号通信システムは、サーバに対してクライアントサーバが正規のサーバであることを自己の電子証明書を用いて代理応答するＣＡ　Ｐｒｏｘｙを備えるので、クライアントの個人情報がサーバに公開されることなく、サーバとクライアントサーバとの間で相互に認証し、暗号通信を行なうことができる。

【図面の簡単な説明】

【図１】　本発明の実施の形態による暗号通信システムの概略ブロック図である。

【図２】　図１に示すサーバ、クライアントサーバ、およびＣＡ　Ｐｒｏｘｙの機能ブロック図である。。

【図３】　電子証明書の構成を示す概略ブロック図である。

【図４】　サーバとクライアントサーバとの間のセッション確立時の第１のフローチャートである。

【図５】　サーバとクライアントサーバとの間のセッション確立時の第２のフローチャートである。

【図６】　サーバとクライアントサーバとの間のセッション確立時の第３のフローチャートである。

【図７】　サーバとクライアントサーバとの間の他の方

24

法によるセッション確立時の第１のフローチャートである。

【図８】　サーバとクライアントサーバとの間の他の方法によるセッション確立時の第２のフローチャートである。

【図９】　サーバとクライアントサーバとの間の他の方法によるセッション確立時の第３のフローチャートである。

【図１０】　ＯＳＩ参照モデルを用いたコンピュータ間の通信するための概略図である。

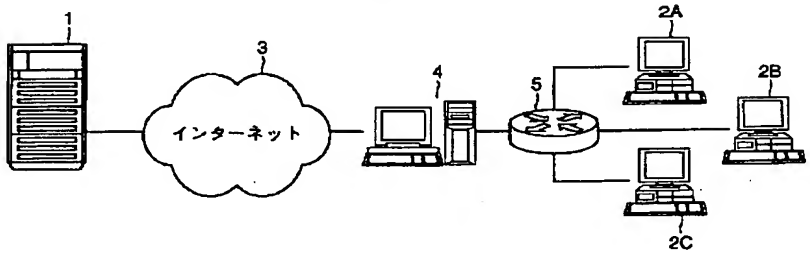
【図１１】　ＯＳＩ参照モデルの各層の機能を説明するための図表である。

【図１２】　従来のＳＳＬ代理応答システムの概略ブロック図である。

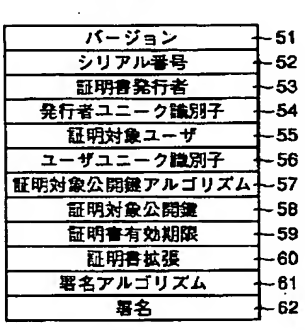
【符号の説明】

１，２４０　サーバ、２Ａ，２Ｂ，２Ｃ　クライアントサーバ、３　インターネット網、４　ＣＡ　Ｐｒｏｘｙ、５　結合器、１０　暗号通信システム、１１，２１　Ｒｅｃｏｒｄ　Ｐｒｏｔｏｃｏｌ部、１２，２２　ＨａｎｄｓｈａｋｅＰｒｏｔｏｃｏｌ部、１３，２３　Ｃｈａｎｇｅ　Ｃｉｐｈｅｒ　ＳｐｅｃＰｒｏｔｏｃｏｌ部、１４，２４　Ａｌｅｒｔ　Ｐｒｏｔｏｃｏｌ部、１５，２５　Ａｐｐｌｉｃａｔｉｏｎ　Ｄａｔａ　Ｐｒｏｔｏｃｏｌ部、１６，２６　アプリケーション部、４１　通信プロトコルキャプチャー部、４２　ＣＲＬチェック部、４３　認証部、４４　代理応答部、５０　電子証明書、５１　バージョン、５２　シリアル番号、５３　証明書発行者、５４　発行者ユニーク識別子、５５　証明対象ユーザ、５６　ユーザユニーク識別子、５７　証明対象公開鍵アルゴリズム、５８　証明対象公開鍵、５９　証明書有効期限、６０　証明書拡張、６１　署名アルゴリズム、６２　署名、２００　ＳＳＬ代理応答システム、２１０　インターネット、２２０　ファイアウォール、２３０　代理サーバ。

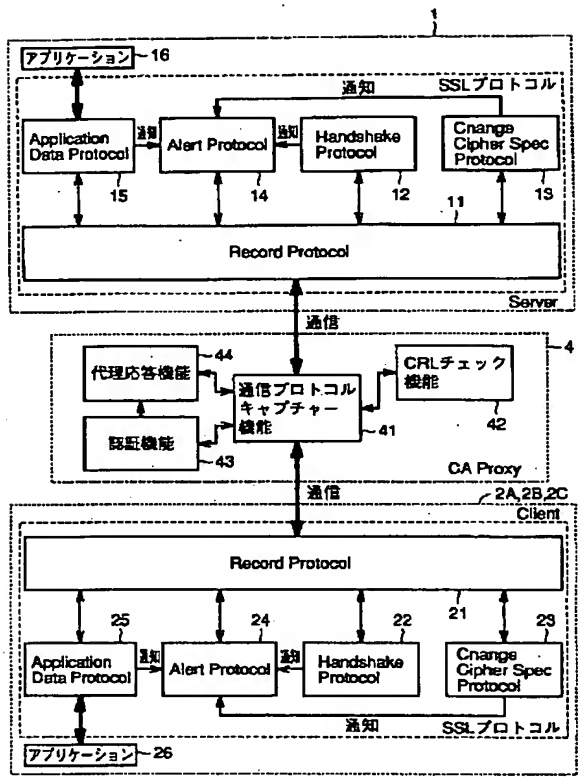
【図１】



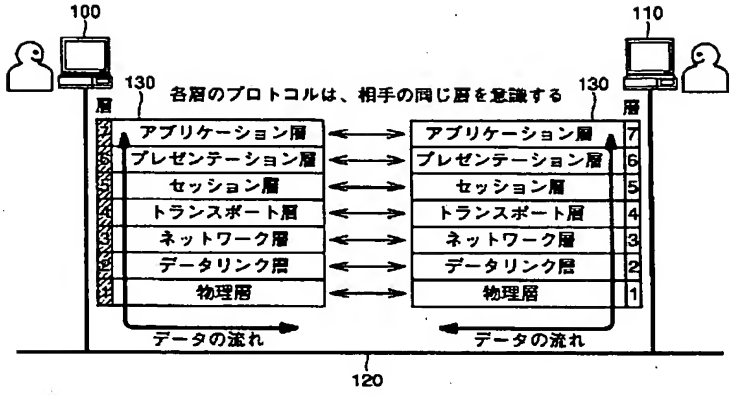
【図３】



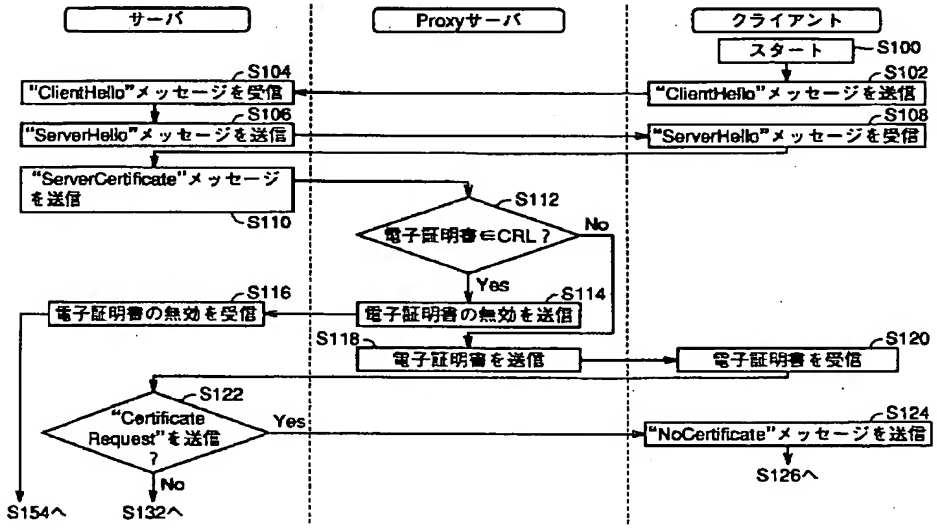
【図２】


1
アプリケーション
16
通知
SSLプロトコル
Application Data Protocol
Alert Protocol
Handshake Protocol
Change Cipher Spec Protocol
15
14
12
11
13
Record Protocol
Server
通信
4
代理応答機能
44
通信プロトコルキャプチャー機能
41
CRLチェック機能
42
認証機能
43
CA Proxy
通信
2A,2B,2C
Client
Record Protocol
25
24
22
21
23
Application Data Protocol
Alert Protocol
Handshake Protocol
Change Cipher Spec Protocol
通知
SSLプロトコル
アプリケーション
26

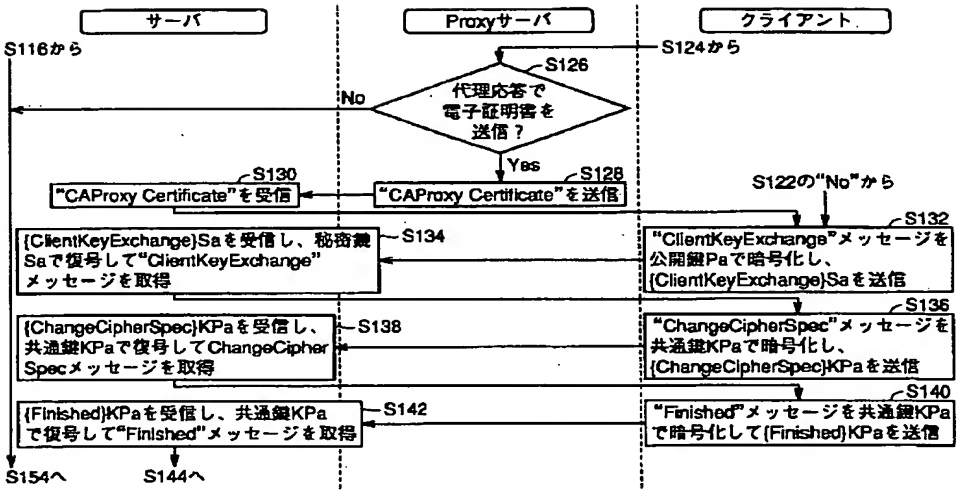
【図１０】


100
110
130
層
各層のプロトコルは、相手の同じ層を意識する
130
層
アプリケーション層
プレゼンテーション層
セッション層
トランスポート層
ネットワーク層
データリンク層
物理層
7
6
5
4
3
2
1
データの流れ
データの流れ
120

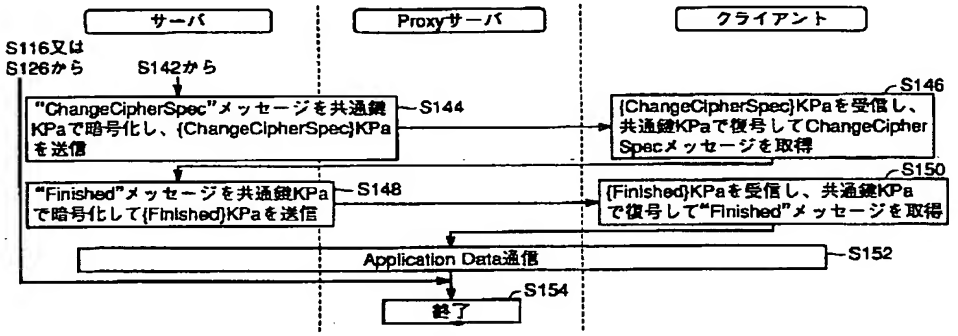
【図４】


サーバ
Proxyサーバ
クライアント
スタート
S100
S102
"ClientHello"メッセージを送信
S104
"ClientHello"メッセージを受信
S106
"ServerHello"メッセージを送信
S108
"ServerHello"メッセージを受信
"ServerCertificate"メッセージを送信
S110
S112
電子証明書∈CRL？
No
Yes
S114
電子証明書の無効を送信
S116
電子証明書の無効を受信
S118
電子証明書を送信
S120
電子証明書を受信
S122
"Certificate Request"を送信？
Yes
No
S124
"NoCertificate"メッセージを送信
S126へ
S154へ
S132へ

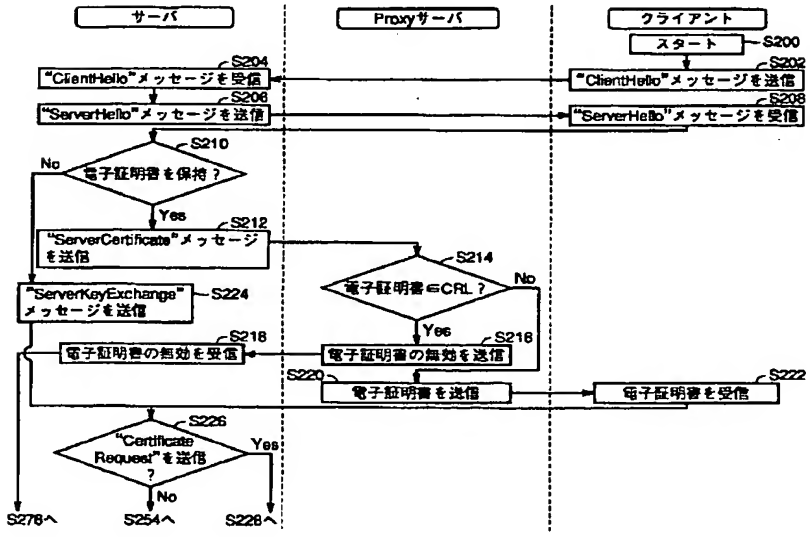
【図５】


サーバ
Proxyサーバ
クライアント
S116から
S124から
S126
代理応答で電子証明書を送信？
No
Yes
S130
"CAProxy Certificate"を受信
S128
"CAProxy Certificate"を送信
S122の"No"から
S132
"ClientKeyExchange"メッセージを公開鍵Paで暗号化し、{ClientKeyExchange}Saを送信
S134
{ClientKeyExchange}Saを受信し、秘密鍵Saで復号して"ClientKeyExchange"メッセージを取得
S136
"ChangeCipherSpec"メッセージを共通鍵KPaで暗号化し、{ChangeCipherSpec}KPaを送信
S138
{ChangeCipherSpec}KPaを受信し、共通鍵KPaで復号してChangeCipher Specメッセージを取得
S140
"Finished"メッセージを共通鍵KPaで暗号化して{Finished}KPaを送信
S142
{Finished}KPaを受信し、共通鍵KPaで復号して"Finished"メッセージを取得
S154へ
S144へ

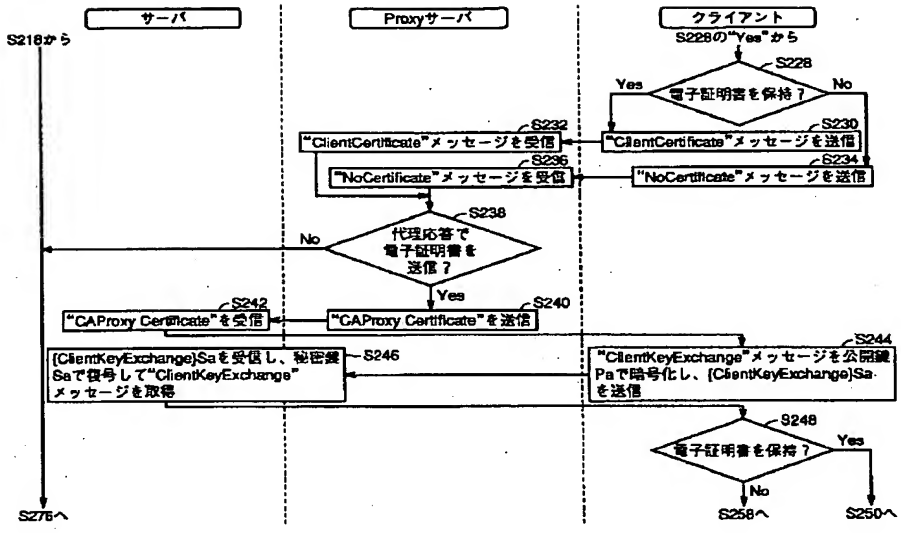
【図６】


サーバ
Proxyサーバ
クライアント
S116又はS126から
S142から
"ChangeCipherSpec"メッセージを共通鍵KPaで暗号化し、{ChangeCipherSpec}KPaを送信
S144
S146
{ChangeCipherSpec}KPaを受信し、共通鍵KPaで復号してChangeCipher Specメッセージを取得
"Finished"メッセージを共通鍵KPaで暗号化して{Finished}KPaを送信
S148
S150
{Finished}KPaを受信し、共通鍵KPaで復号して"Finished"メッセージを取得
Application Data通信
S152
S154
終了

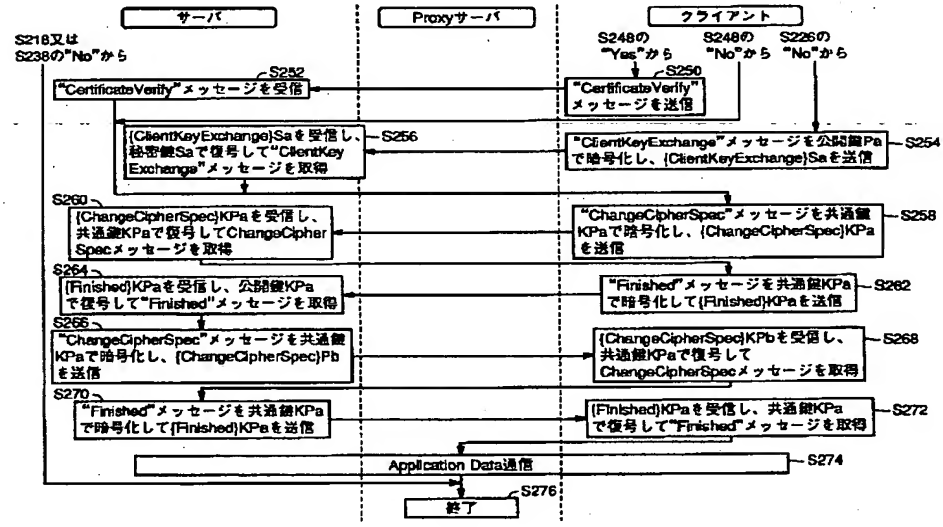
【図７】


サーバ
Proxyサーバ
クライアント
スタート
S200
S202
"ClientHello"メッセージを送信
S204
"ClientHello"メッセージを受信
S206
"ServerHello"メッセージを送信
S208
"ServerHello"メッセージを受信
S210
No
電子証明書を保持？
Yes
S212
"ServerCertificate"メッセージを送信
S214
電子証明書∈CRL？
No
Yes
"ServerKeyExchange"メッセージを送信
S224
S218
電子証明書の無効を受信
S218
電子証明書の無効を送信
S220
電子証明書を送信
S222
電子証明書を受信
S226
"Certificate Request"を送信？
Yes
No
S276へ
S254へ
S228へ

【図８】



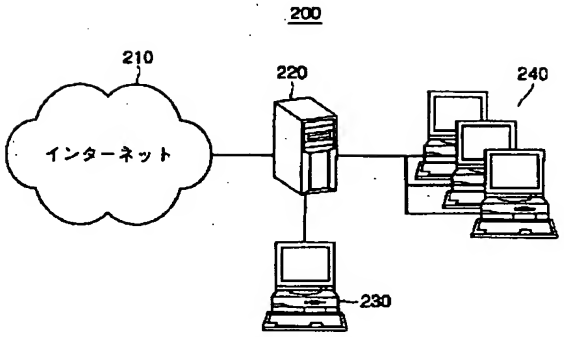
【図９】



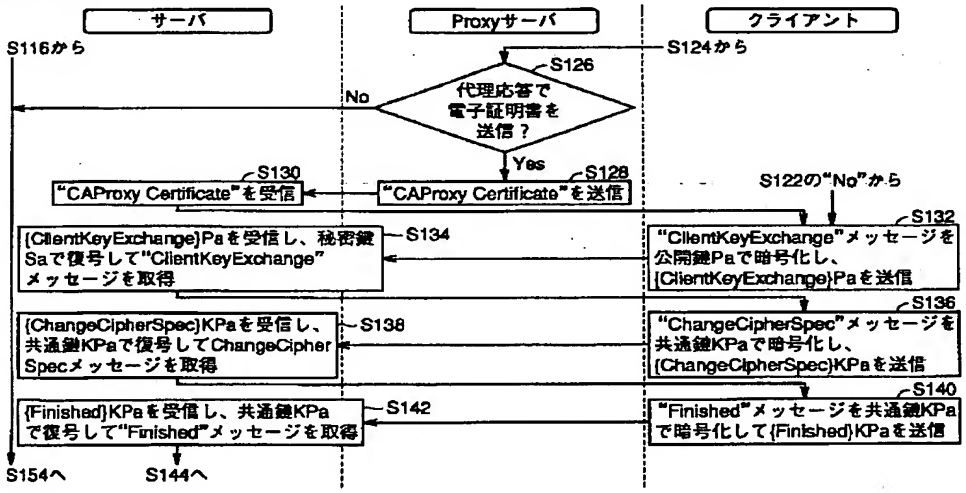
【図１１】

OSI参照モデル上の分類			機能概要
	層	層名称	
上位層	第7層	アプリケーション層 (application layer)	ファイル転送やメッセージ通信(E-mail)など、ユーザが実行する多くのサービス間プロトコルを制御
	第6層	プレゼンテーション層 (presentation layer)	文字コードや画像データの表現形式を制御し、プロセス間におけるデータ形式などを確認する
	第5層	セッション層 (session layer)	アプリケーションプロセス間の情報の流れなど、通信モードの管理や情報転送に関する通信制御
下位層	第4層	トランスポート層 (transport layer)	通信情報の質を高めるための通信制御などを行う。データに抜けがあった場合、相手に通知する
	第3層	ネットワーク層 (network layer)	複数のネットワークにまたがったコンピュータ間のデータ転送やデータの中継機能など
	第2層	データリンク層 (data link layer)	ノード間で信頼性の高いデータ伝送を保証。中継局間のデータ伝送を確実に行う
	第1層	物理層 (physical layer)	データを電気信号に変換し、実際の伝送を行う

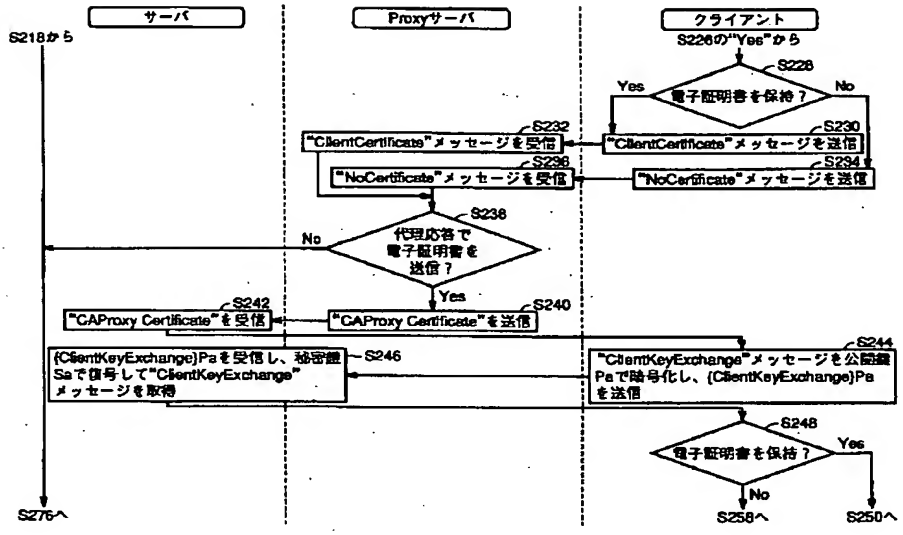
【図１２】



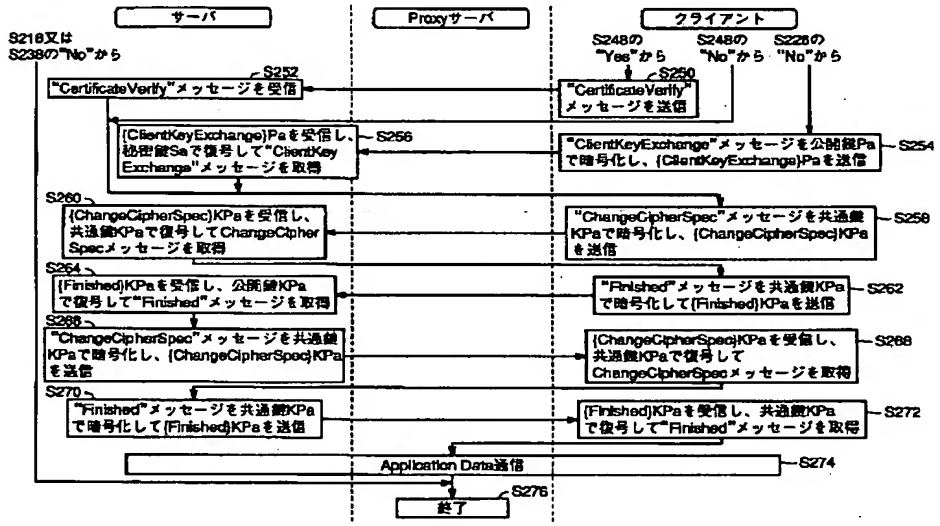
【手続補正書】
【提出日】平成１３年２月２７日（２００１．２．２７）
【手続補正１】
【補正対象書類名】図面
【補正対象項目名】図５
【補正方法】変更
【補正内容】
【図５】



【手続補正２】
【補正対象書類名】図面
【補正対象項目名】図８
【補正方法】変更
【補正内容】
【図８】



【手続補正３】
【補正対象書類名】図面
【補正対象項目名】図９
【補正方法】変更
【補正内容】
【図９】



フロントページの続き

(72)発明者　山崎　達也
京都府相楽郡精華町光台二丁目２番地２
株式会社エイ・ティ・アール環境適応通信
研究所内

Ｆターム(参考)　5J104 AA01 AA07 EA05 JA21 KA02
MA01 NA02 PA07